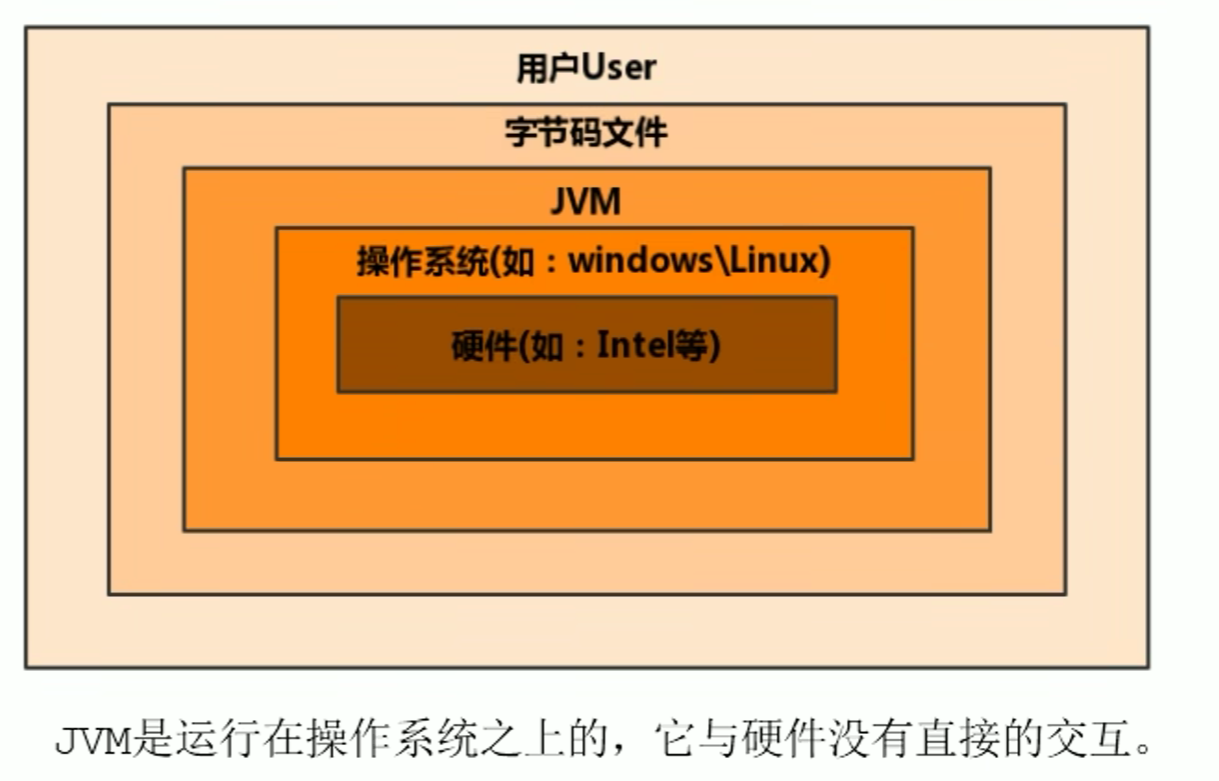
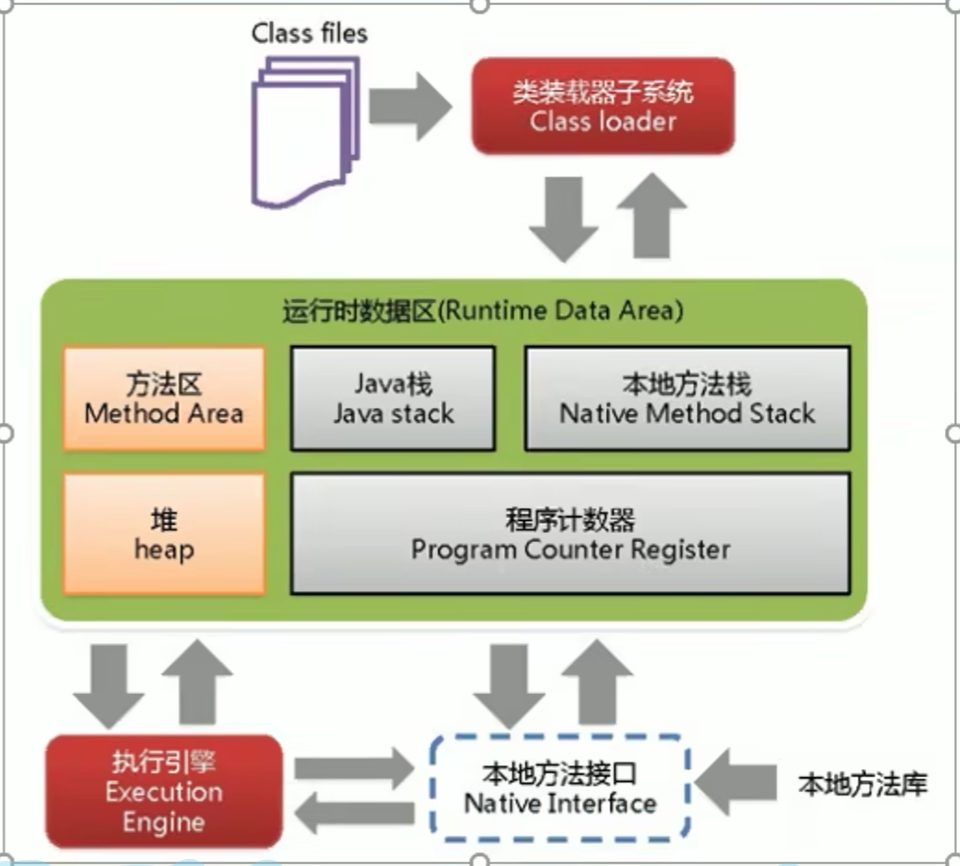
# JVM

1. JVM基本信息
2. 结构说明：

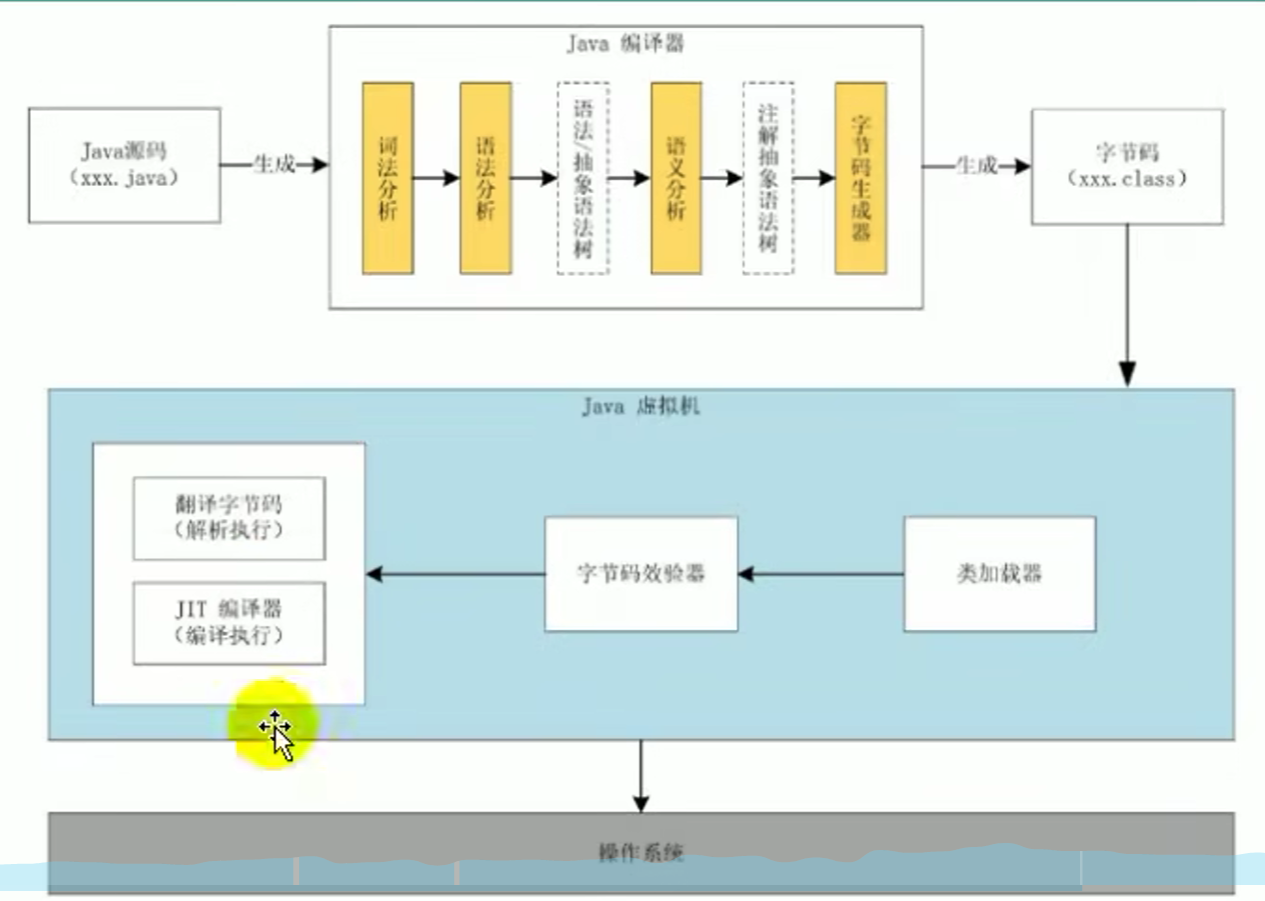
特点：一次编译，处处运行；自动内存管理；自动垃圾回收管理；



JVM整体结构：



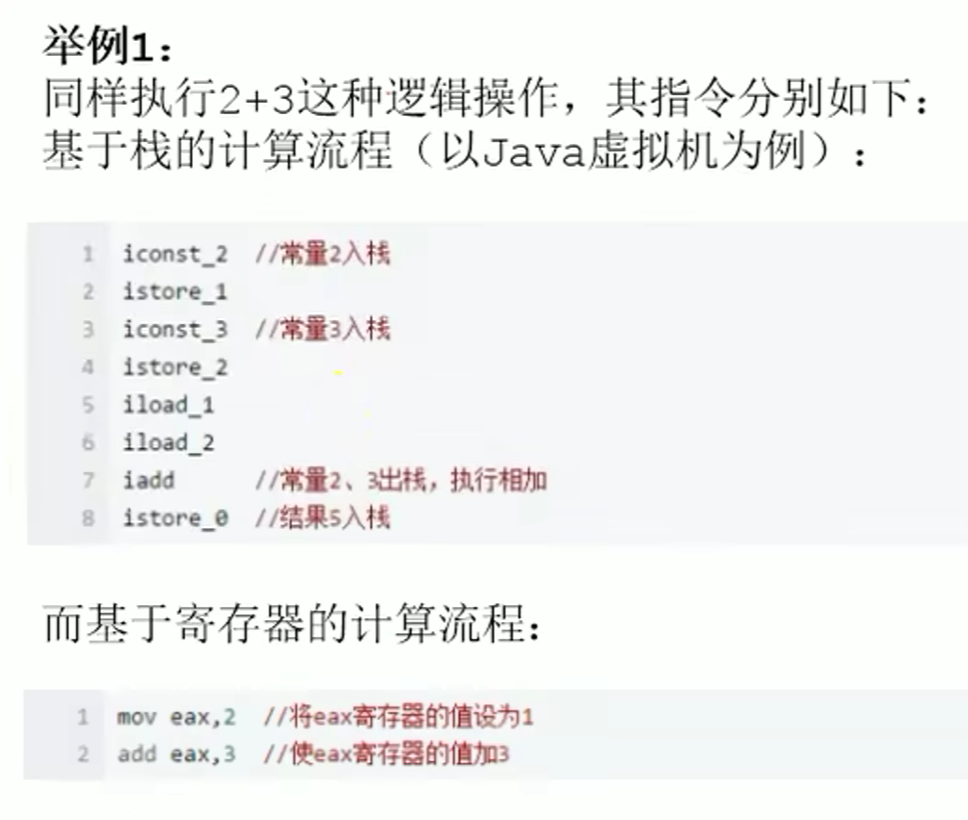
Java代码执行过程：



1. 架构模型：

Java编译器输入的指令流基本上是一种基于栈的指令集架构，另外一种指令集架构是基于寄存器（CPU）的指令集架构。

* 基于寄存器架构的特点：
* 典型的应用是X86的二进制指令集（传统的PC、Android、Davlik虚拟机）；
* 指令集架构完全依赖于硬件，可移植性差；
* 性能优秀和执行更高效；
* 花费跟少的指令去完成一项操作；
* 在大部分情况下，基于寄存器架构的指令集往往都以一地址（一个地址一个操作数）指令、二地址（二个地址一个操作数）指令、三地址（三个地址一个操作数）指令为主，而基于栈式架构的指令集却是以零地址（没有地址只有操作数）指令为主。
* 基于栈式架构的特点：
* 设计和实现更简单，适用于资源受限的系统；
* 避开了寄存器的分配难题（使用零地址指令方式分配）；
* 指令流中的指令大部分是零地址指令，其执行过程依赖于操作栈。指令集更小，编译器更容易实现；
* 不需要硬件支持，可移植性更好，更好实现跨平台。



1. JVM的生命周期：
2. 虚拟机启动：

Java虚拟机的启动是通过引导类加载器（bootstrap class loader）创建一个初始类（initial class）来完成的，这个类是由虚拟机的具体实现来指定的。

1. 虚拟机执行：

* 一个运行中的Java虚拟机有着一个清晰的任务：执行Java程序；
* 程序开始执行时他才运行，程序结束时它就停止；
* 执行一个所谓的Java程序的时候，真真正正在执行的是一个叫做Java虚拟机的进程。

1. 虚拟机退出：

* 程序正常执行结束；
* 程序在执行过程中遇到了异常或错误而异常终止；
* 由于操作系统出现错误而导致Java虚拟机进程终止；
* 某线程调用Runtime类或System类的exit方法，或Runtime类的halt方法，并且Java安全管理器也允许这次exit或halt操作。

1. JVM发展历程：

* Sun Classic VM

Java1.0版本虚拟机，世界上第一款商用Java虚拟机。只提供解释器，不能和JIT编译器共用，JDK1.4时被完全取代。Hotspot内置了此虚拟机。

* Exact VM

Jdk1.2时，sun提供了此虚拟机。可以知道内存中某个位置的数据具体是什么类型，具备热点探测、编译器与解释器混合工作模式。英雄气短，还没推广就被Hotspot取代了。

* Hotspot VM
* Jdk1.3成为默认虚拟机，目前占有绝对的市场地位，称霸武林；
* 通过计数器找到最具编译价值的代码，触发即时编译或栈上替换；
* 通过编译器与解释器协同工作，在最优化的程序响应时间与最佳执行性能中取得平衡。
* JRockit VM

专注于服务器端应用，目前三大主流商用虚拟机之一。不关注程序启动速度，内部不包含解析器实现，全部代码都靠即时编译器编译后执行。是世界上最快的JVM。

* J9 VM

市场定位与Hotspot接近，也是目前三大主流商用虚拟机之一。

* Taobao VM
* 由阿里巴巴的AliJVM团队发布；
* 基于OpenJDK Hotspot VM发布的国内第一个优化、深度定制且开源的高性能服务器版Java虚拟机；
* 创新的GCIH（GC invisible heap）技术实现了off-heap，即将生命周期较长的java对象从heap中移除到heap之外，并且GC不能管理GCIH内部的Java对像，以此达到降低GC的回收频率和提升GC的回收效率的目的；
* GCIH中的对象还能够在多个Java虚拟机进程中实现共享；
* Taobao VM严重依赖intel的CPU，损失了兼容性，但提高了性能；
* 目前淘宝、天猫上线已经把Oracle官方的JVM版本全部替换了。